



Joonas Heikkinen

TIETOKANTASOVELLUKSEN TOTEUTUS ACCESSILLA

TIETOKANTASOVELLUKSEN TOTEUTUS ACCESSILLA

Joonas Heikkinen
Opinnäytetyö
Kevät 2013
Tietojenkäsittelyn koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu

Tietojenkäsittelyn koulutusohjelma

Tekijä: Joonas Heikkinen

Opinnäytetyön nimi: Tietokantasovelluksen toteutus Accessilla

Työn ohjaaja: Sinikka Viinikka

Työn valmistuslukukausi –ja vuosi: Kevät 2013

Sivumäärä: 30 + 6 liitesivua

Opinnäytetyöni toimeksianto tuli Oulun seudun ammattikorkeakoulun Liiketalouden yksiköstä Tietojenkäsittelyn koulutusohjelman puolelta. Työn tavoitteena oli toteuttaa tietokantasovellus Access 2010 –ohjelmalla opetuskäyttöön Tietokantojen perusteet –kurssille keväälle 2012. Kurssilla opiskelijat suorittavat sovelluksella erilaisia oppimistehtäviä, joiden on tarkoitus kasvattaa heidän ymmärrystään tietokantojen toiminnasta ja merkityksestä yritysmaailmassa.

Opinnäytetyössäni käsittelen tietojärjestelmiä, tietokantoja ja tietokannan hallintajärjestelmiä nivoen nämä käsitteet ymmärrettäväksi kokonaisuudeksi ja keskittyen relaatiotietokantaohjelma Accessiin, jolla tietokantasovelluksen toteutin. Lopuksi esittelen toteutetun työn ja kerron siihen johtavasta työskentelystäni.

Asiasanat: Access, tietokanta, tietojärjestelmä, tietokannan hallintajärjestelmä

ABSTRACT

Oulu University of Applied Sciences

Tietojenkäsittelyn koulutusohjelma

Degree Programme in Business Information Systems

Author: Joonas Heikkinen

Title of thesis: The creation of database application using Access.

Supervisor: Sinikka Viinikka

Term –and year when the thesis was submitted: Spring 2013

The Number of Pages: 30 + 6

This thesis was conducted for the department of Business Information Systems in Oulu University of Applied Sciences. The aim of this thesis was to produce a database application using Microsoft Access 2010 – a relational database management system. The application was to be used for educational purposes in a course which deals with databases and their significance in enterprise use. The course was held in spring 2012 and students performed different tasks with the application aimed to increase their knowledge of databases.

In my thesis I deal with the subjects of information system, database and database management system explaining them and bringing the subjects closer to each other to form the big picture. However my main focus will be on the relational database management system Access 2010. Finally I will present the application which I have created and tell of the process of creating it.

Keywords: Access, database, database management system, information system.

SISÄLLYS

1	TOIMEKSIANTAJA JA TOIMEKSIANTO	6
2	TIETOKANTA TIETOJÄRJESTELMÄN OSANA.....	9
2.1	Tietojärjestelmä	9
2.2	Tietojärjestelmän merkitys organisaation toiminnassa	10
2.3	Tietokanta ja tietokantasovellus	11
2.4	Relaatiotietokanta.....	12
3	MICROSOFT ACCESS	16
3.1	Access-objektit	16
3.2	Accessin edut.....	17
3.3	Muita tietokannan hallintajärjestelmiä	18
4	DEMOX2 JA SEN TOTEUTUS	20
4.1	Alkuvalmistelut ja tietokanta	20
4.2	Lomakkeet ja SQL-kyselyt	20
4.3	Raportit.....	23
4.4	Visual Basic for Applications	24
4.5	Viimeistely	26
5	YHTEENVETO JA POHDINTA	27
	LÄHTEET.....	29

1 TOIMEKSIANTAJA JA TOIMEKSIANTO

Opinnäytetyöni toimeksianto tuli Oulun ammattikorkeakoulun Liiketalouden yksiköstä Tietojenkäsittelyn koulutusohjelman puolelta. Tietojenkäsittelyä opiskellaan Oulussa Liiketalouden yksikössä ja tutkintonimikkeenä on tradenomi, joka on alempi korkeakoulututkinto. Tietojenkäsittelyn tradenomit saavat laaja-alaiset tiedot ICT-alasta sekä tietojen soveltamisesta käytäntöön. Oulussa opiskelijat voivat erikoistua esimerkiksi järjestelmäasiantuntemukseen tai web-sovelluskehitykseen. Keskeisiä Tietojenkäsittelyn koulutusohjelman opintoja ovat tietojärjestelmien suunnittelu ja ohjelmointi, käyttöliittymät, digitaalinen media, web- ja mobiilisovellusten kehittäminen, tietokonejärjestelmät ja tietoliikenne. (OAMK - Luonnontieteiden ala (tietojenkäsittely), hakupäivä 4.11.2012.)

Toimeksiantaja tarvitsi tietokantasovellusta lähiopetuksen tueksi Tietokantojen perusteet – kurssille keväälle 2012. Kurssilla tutustutaan tietokantoihin ja opetetaan tietokantojen merkitystä informaatiojärjestelmässä sekä yrityksen toiminnassa. Tietokantasovelluksen tuli tarjota opiskelijoille eräänlainen simulaatio kuvitteellisen pienehkön postimyyntiyrityksen yritystietokantajärjestelmän toimintaan. Opiskelijat pääsevät sovelluksen avulla suorittamaan monia erilaisia oppimistehtäviä, jotka pyrkivät mallintamaan realistisia yritystietokantaohjelmiston käyttötapauksia yrityksen työntekijän näkökulmasta.

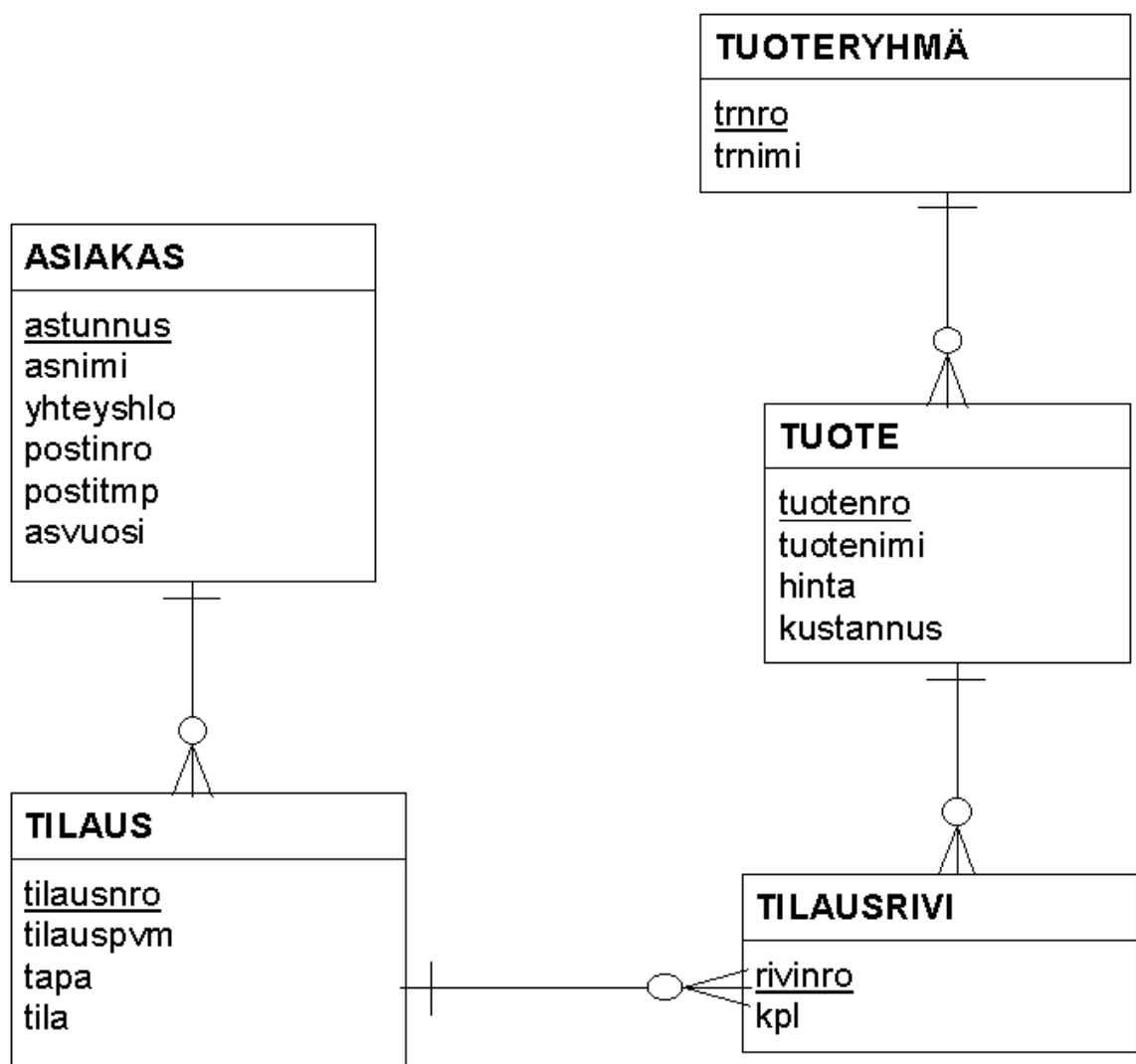
Toimeksiantajan kanssa päätimme yhteistuumin, että sovellus toteutetaan Microsoft Office – ohjelmistopakettiin kuuluvalla Microsoft Access 2010 tietokantojen hallintaohjelmalla. Access-tietokanta toimii ilman erillistä tietokantapalvelinta, ja muiden Office-tuotteiden tapaan, Accessilla toteutettu sovellus luo tietokoneelle vain yhden tiedoston. Nämä seikat mahdollistavat sovelluksen varsin helpon käyttöönoton oppitunnilla. Oppitunnilla opiskelijat saavat työasemilleen kopion Access-sovelluksen sisältävästä tiedostosta, jolla pystyvät kukin suorittamaan omatoimisesti ennaltamäärätyt oppimistehtävät.

Toimeksiantajalla oli varsin selkeä kuva siitä, mitä sovelluksen tuli pitää sisällään, ja mitä sillä pitää pystyä tekemään. Toiminnallisena vaatimuksena toimivatkin toimeksiantajan laatimat alustavat sanalliset oppimistehtävät, jotka opiskelijoiden tulee pystyä sovelluksella toteuttamaan; toiminnot, jotka sovelluksella tulee pystyä suorittamaan; sekä suunnitelma tietokannan sisällöstä.

Yhdessä nämä sekä rajaavat sovellusta paisumiselta, että asettavat toiminnalliset minimivaatimukset.

Tietokanta ja toiminnot

Toimeksiantaja on suunnitellut tietokannan rakenteen ja sen pohjana on toimeksiantajan virtuaaliammattikorkeakoululle suunnittelema opetustietokanta Demox, joka niin ikään mallintaa kuvitteellisen postimyyntiyrityksen yritystietokannan toimintaa. Kuvassa 1 on esillä Demox – tietokannan tietokantarakenne tauluineen, sarakkeineen ja taulujen välille muodostettuine yhteyksineen.



Kuva 1. Demox-opetuskannan käsitemalli

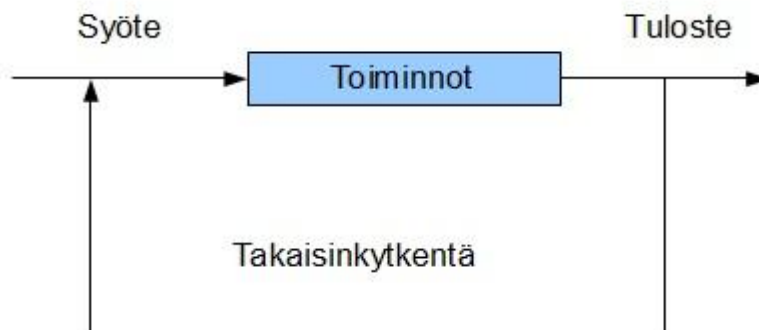
Alkuperäisen vaatimusmäärittelyn mukaan, toteutettavan sovelluksen kanta eroaa yllä esitetystä käsitemallista muun muassa kahdella uudella taululla ja muutamalla poistuvalla sarakkeella.

Sovellukseen määriteltiin seuraavanlaiset toiminnot, jotka toteutettavalla tietokantasovelluksella tuli kyetä suorittamaan. Toiminnot olivat tuoteryhmien, tuotteiden, hintojen, asiakastietojen, tilausten ja toimitusten ylläpito, mahdollisuus lisätä tietokantaan uusi postinumero ja postitoimipaikka sekä kattava raportointi tuotteista, asiakkaista ja tilauksista.

2 TIETOKANTA TIETOJÄRJESTELMÄN OSANA

2.1 Tietojärjestelmä

Tietojärjestelmän määritelmää voidaan lähteä avaamaan järjestelmän käsitteestä. Mitä tahansa järjestelmää voidaan luonnehtia joukkona keskinäisissä suhteissa olevia osasia, jotka ovat vuorovaikutuksessa toistensa kanssa muodostaen kokonaisuuden, ja jotka työskentelevät kohti yhteistä päämäärää. Järjestelmän pääasialliset osat ovat syöte, toiminnot, tuloste ja kontrolloiva takaisinkytkentä. Syöte menee järjestelmään sisään, tuloste tulee ulos ja toiminnot mahdollistavat syötteen muuntamisen tulosteeksi. Takaisinkytkentä mahdollistaa tulosteen tuomisen takaisin järjestelmään ja taas syötteeksi. Tietojärjestelmässä syötteet ovat joko järjestelmän ulkopuolisesta toimintaympäristöstä tulevaa tai jo tietokannoissa olevaa dataa, tai tietokantoihin tehtäviä kyselyitä, jotka operoivat tietokantoihin tallennettua dataa. (Laakkonen ym. 2011, 82-83.)



Kuva 2. Tietojärjestelmä (Laakkonen ym. 2011, 83)

Olennaisena osana tietojärjestelmää on myös sen toimintaympäristö eli käyttäjät. Kaiken kaikkiaan tietokoneperustainen tietojärjestelmä sisältää järjestelmän tietokonelaitteet eli palvelimet, päätteet, tulostimet ynnä muut järjestelmän toiminnalle tarpeelliset atk-laitteet. Toiseksi ohjelmiston, joka koostuu järjestelmän ohjelmasta, joka kontrolloi laitteistoa (palvelinohjelma) sekä ohjelmasta, joka tarvitaan halutun informaation tuottamiseen

(tietokannanhallintaohjelma esim. Access.) Kolmantena tulevat järjestelmän käyttäjät, jotka hallinnoivat järjestelmän syötteitä, tulosteita, ohjelmistoa ja laitteistoa. (Laakkonen ym. 2011, 84.)

2.2 Tietojärjestelmän merkitys organisaation toiminnassa

Tietojärjestelmät ovat nykypäivänä merkittävä osa lähes minkä tahansa yrityksen toimintaa. Yritykset hyödyntävät enemmän ja enemmän tietotekniikkaa, jopa niin paljon, että osa yrityksistä on täysin riippuvaisia tietojärjestelmistään. (Ruohonen ym. 2005, 7.) Tietotekniikka ei siis ole enää pelkkä toimintaa tehostava työkalu, vaan peruspilari, jonka päälle koko yrityksen liiketoiminta rakentuu ja pilarin pettäessä koko liiketoiminta vaarantuu. Tietojärjestelmätieteen tutkijan Hannu Salmelan mukaan tietojärjestelmiin ja niiden ylläpitämiseen ei osata tai haluta panostaa riittävästi, joka taas johtuu siitä, että tietojärjestelmien merkitys on kasvanut vähitellen niin suurin mittasuhteisiin, että sitä on joskus vaikea ymmärtää. Salmelan mukaan tietotekniikkaa ja tietojärjestelmiä pidetäänkin liian usein vain ylimääräisinä kuluerinä, vaikka IT-investoinnit tulisi nähdä välttämättöminä investointeina toiminnan ylläpitämiseen ja kehittämiseen. (Turun kauppakorkeakoulun sidosryhmälehti Mercurius, hakupäivä 28.10.2012.)

Yrityksissä on monia erilaisia tietojärjestelmiä, joilla yleisesti ottaen hoidetaan yrityksen tietovirtojen ohjausta ja hyväksikäyttöä. Yritysten eri toimintayksiköillä, tai osastoilla, on juuri sille osastolle ominaisia tietojenkäsittelytarpeita ja tämän takia järjestelmiä on monesti useita. Yksittäinen tietojärjestelmä voi myös olla osa suurempaa kokonaisuutta ja täten toimia emojärjestelmän osajärjestelmänä. (Ruohonen ym. 2005, 33.)

Yrityksen toimintojen eri osa-alueet voivat todellakin olla raskaasti tietojärjestelmien kyllästämiä. Esimerkiksi tuotantoyrityksen toiminnot ja niiden toimintokohtaiset tietojärjestelmät voidaan jakaa valmistuksen, markkinoinnin, kuljetuksen ja jakelun, tukitoimintojen, taloushallinnon, henkilöstöhallinnon, tuotekehityksen ja yleishallinnon ja johtamisen kategorioiden alle. (Ruohonen ym. 2005, 34.)

Tuotantoyrityksessä valmistuksen tietojärjestelmät ovat tietojärjestelmien ydin. Valmistuksen tietojärjestelmät tukevat valmistuksen toimintoja, kuten hankintaa, vastaanottoa, laadunvalvontaa, varastonhallintaa sekä materiaaliostojen, kapasiteetin, tuotannon ja tuotantorakenteen suunnittelua. (Ruohonen ym. 2005, 34-35.)

Markkinoinnin tietojärjestelmät tukevat yrityksen myyntiin ja asiakashallintaan liittyviä toimintoja monipuolisilla asiakastietokannoilla, asiakkuuksien seuranta –ja hyödyntämisjärjestelmillä ja erilaisilla myynnin tehostamisjärjestelmillä. (Ruohonen ym. 2005, 35-36.) Kuljetuksen ja jakelun tietojärjestelmillä tavoitellaan asiakkaille lisäarvoa tuottavia palveluita ja vähentyneitä logistiikkakustannuksia. Tällaisia palveluita voivat olla esimerkiksi lähetetyn paketin seuranta ja reititys. (Ruohonen ym. 2005, 37-38.)

Taloushallinnon järjestelmät huolehtivat yrityksen rahavirtojen ohjauksesta ja tukevat toimintoja kuten kustannuslaskenta, kirjanpito ja raportointi. Taloushallinnon ja henkilöstöhallinnon yhteisenä tietojärjestelmänä voidaan nähdä palkanlaskennan tietojärjestelmä. Muita henkilöstöhallinnon tietojärjestelmiä ovat mm. rekrytoinnin, kulunvalvonnan ja työaikojen seurannan tietojärjestelmät. (Ruohonen ym. 2005, 38-40.) Edellä esitettyjen esimerkkien valossa on helppo nähdä, miten oleellisessa osassa tietojärjestelmät ovat yritysten toimintaa.

2.3 Tietokanta ja tietokantasovellus

Perusolemukseltaan tietokanta voidaan nähdä kokoelmana tietoa, jota säilytetään pitkä aika. Yleensä tietokannasta puhuttaessa tarkoitetaan kuitenkin tietokokoelmaa, jota hallitaan jollakin tietokannan hallintajärjestelmällä. Tietokannan hallintajärjestelmällä käyttäjä voi luoda uusia tietokantoja ja hakea ja muokata tietokannassa olevaa tietoa. Järjestelmä myös huolehtii siitä, että tietokannan tiedot ovat turvassa ulkopuolisilta, ja että tiedot säilyvät eheänä pitkiä aikoja. Perusvaatimus tietokannan hallintajärjestelmälle on myös yhteiskäyttö, joka tarkoittaa että tietokantaa voi operoida samanaikaisesti usea henkilö, ilman että tämä aiheuttaa minkäänlaisia ongelmia. Ensimmäiset kaupalliset tietokannahallintajärjestelmät tulivat markkinoille 1960-luvun loppupuolella. Silloiset tietokannat ja niiden hallintajärjestelmät olivat tyypiltään hierarkkisia tai verkkomallisia. Nykyään näillä malleilla on enää lähinnä historiallista arvoa relaatiomallin syrjäytettyä ne 1970-luvulla. (Ullman ym. 1997, 1-2.)

Kun tässä työssä puhutaan tietokantasovelluksesta, tarkoitetaan sillä tietokannan hallintajärjestelmällä toteutettua käyttöliittymää tietokannan hallintaan. Sovelluksen on tarkoitus helpottaa tietokannan päivittäiseen käyttöön liittyviä toimintoja, kuten tietokannan tietojen tarkastelu, lisäys, poistaminen ja päivittäminen. Lambertin (2008, 289) mukaan tietokanta muuttuu tietokantasovellukseksi, kun siihen lisätään käyttöä helpottavia työkaluja kuten kyselyjä, lomakkeita, raportteja, makroja ja moduuleita.

2.4 Relaatietietokanta

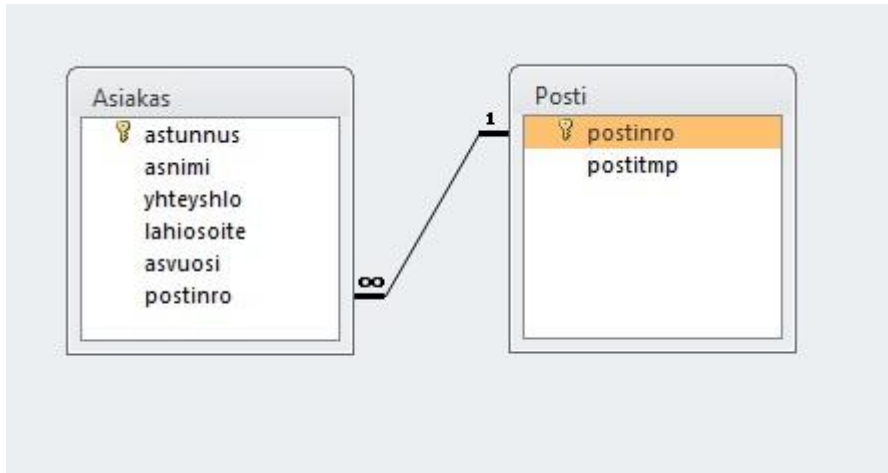
Tietokantojen maailma koki mullistuksen vuonna 1970, kun E. F. Codd julkaisi artikkelinsa A Relational Model of Data for Large Shared Data Banks, jossa hän esittelee relaatiotietokantamallin, joka on sittemmin syrjäyttänyt aiemmin käytetyt hierarkkisen ja verkkomallisen tietokantatyypin. Käytännössä lähes kaikki uudet tietojärjestelmät tehdään relaatiotietokantapohjaisiksi. (Hovi ym. 2005, 2.)

Coddin mallissa keskeisin ajatus on säilyttää tieto eheänä sekä riippumattomana tietokannan hallintajärjestelmistä, eli ohjelmista, joilla tietokantaa käytetään. Coddin aikaiset tietojärjestelmät – ja mallit olivat monesti hyvin paljon sidoksissa tietokannan hallintajärjestelmiin ja järjestelmät tekivät mm. indeksointiin, saantipolkuihin ja tiedon esitystapaan liittyviä merkintöjä tietokantaan, jolloin tiedon riippumattomuus vaaraantui ja järjestelmä tuli alttiiksi virheille (Codd 1970, 377-378). Tietoriippumattomuuden ansiosta tietokannan hallintajärjestelmän käyttäjän ei tarvitse tietää tiedon fyysisistä rakenteista tietokannassa päästäkseen käsiksi tietokannan tietoihin. Käyttäjälle riittää, että hän osaa käyttää tietokannanhallintajärjestelmänä toimivaa ohjelmistoa. (Hovi ym. 2005, 12-13.)

Hovi ym. (2005, 8) jakaa relaatiomallin kolmeen osaan: **rakenne**, **käsittely** ja **eheyssäännöt**. Relaatiomallin rakenne koostuu useista tauluista, joita Codd nimitti alkuperäisessä mallissaan relaatioiksi. Taulun vaakariviä kutsutaan riviksi, tietueeksi tai monikoksi (Coddilla tuple eli monikko.) Monikon eri pystysarakkeet ovat nimeltään sarakkeita, attribuutteja tai kenttiä. (Codd 1970, 377.) Tietokannan tiedot jaetaan useisiin tauluihin, joiden välille muodostetaan yhteyksiä. Tällä halutaan relaatiomallissa välttää redundanssia, joka tarkoittaa tiedon toistamista. (Codd 1970, 383-386.)

Tärkeä osa tietokannan suunnitteluprosessia on perusavaimen määrittäminen. Perusavaimeksi valitaan jokin taulun sarake. Jokaiselle taululle täytyy valita perusavain, eli uniikki arvo, jota ei ole millään muulla tietueella. Henkilötietoja sisältävässä taulussa esimerkiksi henkilötunnus voisi olla perusavain. (Hovi ym. 2005, 9.) Viiteavainta käyttämällä muodostetaan yhteyksiä taulujen välille. Taulusta viitataan toisen taulun perusavaimeen, jolloin viittaava sarake on viiteavain. Viittaavaa taulua sanotaan lapsitauluksi ja viittauksen kohteena olevaa taulua isätauluksi. Kuvassa 3 näkyy Asiakas ja Posti –taulut ja niiden sarakkeet sekä taulujen välille muodostettu yhteys. Asiakas – taulussa astunnus (asiakastunnus) on perusavain, Posti-taulussa postinro (postinumero.) Asiakas

–tauluun on haluttu lisätä tieto asiakkaan postinumerosta, jolloin on luotu yhteys tauluun, joka hallinnoi postinumeroita. Asiakas –taulun postinro sarake on siis viiteavain ja Asiakas –taulu on tässä tapauksessa lapsitaulu ja Posti isätaulu.



Kuva 3. Yhteys taulujen välillä

Coddin relaatiomallissa tietoja käsitellään joukko-opillisesti. Yhden tai useamman taulun riveihin kohdistetaan joukko-operaatioita, joilla tietokannan tietoja ylläpidetään ja tarkastellaan mm. poistamalla, muokkaamalla, lisäämällä ja hakemalla tietoa. Relaatiotietokantatuotteissa joukko-operaatiot toteutetaan Structured Query Language (SQL) -kielellä, useimmat relaatiotietokantatuotteet ymmärtävät vain SQL-kieltä. (Hovi ym. 2005, 10.) Relaatiomallissa toteutuu tietokannan eheys, kun tiedot ovat oikein, ristiriidattomia ja vastaavat reaali maailmaa (Hovi ym. 2005, 11). Ristiriitoja syntyy, jos esim. sama henkilö on kahdesti henkilötietoja käsittelevässä taulussa, tai henkilöllä on kaksi eri osoitetta eikä tiedetä, kumpi on oikea.

Relaatiomallin alkaessa yleistyä 1980-luvulla ja useiden tahojen alkaessa soveltaa ja kehittää Coddin mallia suuntiin, jotka eivät häntä miellyttäneet, julkaisi Codd Computer World –lehdessä 12 sääntöä (sääntöjä on 13, Coddin laskutapa alkaa nollasta) määrittämään relaatiomallin mukaista tietokantaa. Mikään moderni laajalti levinnyt tietokanta ei täytä täysin Coddin sääntöjen mukaisia vaatimuksia. Hovi ym. (2005, 11-12) haluaa korostaa kirjassaan säännön numero 10 kahta lisämäärittystä, avaineheyttä ja viite-eheyttä. Avaineheyden mukaan perusavaimen arvo ei saa olla NULL eli tyhjä. Jos perusavaimen arvo on tyhjä, siihen ei voi viitata. Mikäli viite-eheys ei toteudu, jää tietokantaan kuolleita viitteitä, jotka eivät osoita minnekään ja viite-eheys särkyy. Usein tietokantaohjelmat valvovat viite-eheyksiä särkymiseltä ja estävätkin viitteen kohteena olevan tietueen poistamisyritykset.

Coddin 12 sääntöä

0. Relaatiotietokannan hallintajärjestelmän tulee pystyä hallitsemaan tietokantaa käyttämällä vain relaatio-ominaisuuksia.
1. Kaikki tieto relaatiotietokannassa esitetään vain yhdellä tavalla - arvoina tauluissa.
2. Kaikki tieto täytyy olla saatavilla ja löydettävissä taulun nimen, perusavaimen ja sarakkeen perusteella.
3. NULL -arvolla kuvataan puuttuvia tietoja systemaattisesti tietotyypistä riippumatta.
4. Tietokannan kuvaukseen täytyy päästä käsiksi samoilla metodeilla, joilla tietokannan muutakin tietoa käsitellään.
5. Järjestelmän täytyy tukea ainakin yhtä relaationaalista kieltä. Kielellä täytyy lisäksi olla lineaarinen syntaksi, sitä pitää voida käyttää interaktiivisesti että ohjelman sisältä ja sillä täytyy pystyä suorittamaan tiedon määrittely, muokkaus, valtuutus, eheys ja siirto-operaatioita.
6. Näkymien täytyy olla päivitettävissä.
7. Tiedon lisäys, päivitys ja poisto-operaatiot täytyy pystyä kohdistamaan useampaan kuin yhteen tietueeseen.
8. Mikäli tiedon fyysinen esitystapa muuttuu, ohjelman jolla tietokantaa käytetään, ei tarvitse muuttua.
9. Mikäli tiedon looginen esitystapa muuttuu, ohjelman jolla tietokantaa käytetään, ei tarvitse muuttua.
10. Tietokannan eheysmäärittelyt täytyy toteuttaa ilman tietokannan hallintaohjelmaa ja niitä täytyy pystyä muuttamaan ilman muutoksia tietokannan hallintaohjelmaan. Lisäksi perusavaimen arvo ei saa olla NULL (avaineheys) ja jokaista viite-avainta kohti täytyy löytyä sitä vastaava perusavain (viite-eheys).
11. Tiedon hajauttaminen ei saa vaikuttaa tiedon käsittelyyn tietokannan käyttäjän näkökulmasta.
12. Alemman tason tiedonkäsittely (yksi tietue kerrallaan) ei voi ohittaa ylemmän tason (useita tietueita kerralla) määrittelyä. (Wikipedia 2012, hakupäivä 20.10.2012.)

SQL

Relaatiotietokantoihin liittyy olennaisena osana SQL. SQL on IBM:n kehittämä standardoitu kyselykieli relaatiotietokannassa olevien tietojen käsittelyyn. Tietokantaa voi operoida SQL:n avulla syöttämällä käyttöliittymästä SQL-lauseita tietokannalle, mutta usein SQL:n toiminnot ovat upotettu myös käytettävän ohjelman käyttöliittymään. Tällöin tietokantaan kyselyitä tekevän käyttäjän ei välttämättä tarvitse osata kirjoittaa varsinaista SQL-kieltä vaan antaa ohjelmalle muuttujat, joiden avulla haluttu operaatio saadaan suoritettua. (Hovi ym. 2005, 10.)

3 MICROSOFT ACCESS

Access on Microsoftin Office –tuoteperheeseen kuuluva tietokannan hallintajärjestelmä, joka on ollut markkinoilla vuodesta 1992. Access soveltuu parhaiten henkilökohtaiseen sekä kevyeen yrityskäyttöön; rajat alkavat tulla vastaan, kun tietokantaan tekee yhtäaikaaisesti muutoksia yli 10 henkilöä, tai tietokannan koko kasvaa kahteen gigatavuun (MacDonald 2010, 6.)

Tietokannan hallintajärjestelmänä Access on erittäin monipuolinen. Accessin monet työkalut ja tukimm. Visual Basic –ohjelmointikielelle mahdollistavat Accessin roolin sovelluskehittimenä. Lisäksi Access tarjoaa oman tietokantamoottorinsa Microsoft JET Database Engine, joka mahdollistaa relaatiotietokantojen luomisen ja hallinnan. Tästäkin syystä Access soveltuu hyvin pienten ja keskikokoisten tietovarastojen hallintaan yhden ohjelman hoitaessa kolmea roolia: tietokantapalvelin, käyttöliittymä tietokannan hallinnointiin, sekä sovellus, jolla tietokannan päivittäinen käyttö tapahtuu.

Accessin käyttämä tiedostomuoto on versiosta 2007 lähtien ollut .accdb –tiedosto, joka on paranneltu versio aiemmasta .mdb –tiedostomuodosta. Accessin tietokantamoottori mahdollistaa kaiken Accessin käsittelemän datan tallentamisen yhteen ja samaan tiedostoon. Myös jaettu tiedostomalli, jossa Access-objektit eli taulut, kyselyt, raportit jne. sijaitsevat eri tiedostossa kuin itse tietokanta, on mahdollinen. Esimerkiksi tietokantatiedosto voidaan ladata palvelimelle ja Access-objektit sisältävä tiedosto käyttäjien koneille. Tämänkaltaisen järjestely on hyödyksi ympäristössä, jossa tietokannalla on useita samanaikaisia käyttäjiä, ja on tarvetta vähentää verkon rasitusta sekä mahdollistaa loppukäyttäjien muokata tietokantasovellusta omien mieltymystensä mukaisiksi.

3.1 Access-objektit

Access tietokanta –ja käyttöliittymä rakennetaan luomalla erilaisia objekteja, jotka kommunikoivat toistensa ja tietokannan tietueiden kanssa. Näistä keskeisimpiä ovat taulut, kyselyt, raportit ja lomakkeet.

Taulu

Accessin tärkein objekti on taulu, sillä niihin varastoidaan tietokannan tiedot. Tiedot esitetään taulussa riveinä ja sarakkeina käsiteltävän tiedon sisällön perusteella. Mikäli taulujen välille muodostetaan yhteyksiä perus –ja viiteavainten avulla, kutsutaan viitattavaa, perusavaimen sisältävää taulua isätauluksi ja viittavaa, viiteavaimen sisältävä taulua, lapsitauluksi.

Kysely

Kyselyt etsivät niille kerrotun kaavan mukaan haluttua tietoa tauluista, jatkokäsittelevät tiedon suorittamalla niillä esimerkiksi laskutoimituksia, päivittävät, järjestelevät tai poistavat taulujen tietoja.

Lomake

Lomakkeet ovat keskiössä rakennettaessa loppukäyttäjälle näkyvää tietokannan käyttöliittymää. Lomakkeen avulla suoritetaan tietokannan tietojen ylläpitoon liittyviä tehtäviä, kuten tietojen päivitys, lisääminen ja poistaminen. Lomakkeen avulla loppukäyttäjä suorittaa kyselyitä esimerkiksi klikkaamalla hiirellä lomakkeelle luotua painiketta.

Raportti

Raportit ovat työkalu tietokannan tietojen esittämiseen tulostusystävällisessä ja esittelykelpoisessa muodossa. Keinosen (2010, 18) mukaan raportti onkin ainoa tyylikäs tapa luoda tuloste. Raportti kokoaa halutut tiedot tietokannasta esimerkiksi kyselyiden avulla ja asettaa ne raportille määriteltujen muotoilujen sisälle. Raportit voivat sisältää myös muun muassa kuvia ja Excel-tyyppisiä kaavioita.

3.2 Accessin edut

Access on osana Office –ohjelmistopakettia erittäin yhteistyökykyinen muiden Office –tuotteiden kanssa ja tiedot liikkuvat vaivatta sovellusten välillä. Esimerkiksi Excel -taulukon pystyy kääntämään suoraan Access –tauluksi. Tietokantaohjelmistoa valittaessa on myös työntekijöille

eduksi Microsoft Office tuntemus, koska myös Access käyttää muista Office –tuotteista tuttua nauha-käyttöliittymää.

Verrattuna esimerkiksi Microsoftin SQL Server –ohjelmistoon, Access ei tarvitse palvelinta – tietokantaa ajetaan vaivattomasti koneelta, jolle Access on asennettu ja tietokannan sisältävä tiedosto kopioitu. Palvelimilla pyörivät tietokannat vaativat yleensä myös erikseen ohjelmiston, jolla kommunikoida palvelimen tietokannan kanssa, Accessissa on itsessään kaikki tietokannan hallinnointiin tarvittavat ominaisuudet. (MacDonald 2012, 5.)

Access –osaaminen ei vaadi ohjelmointikielten opettelua edistyneen graafisen käyttöliittymän ansiosta, vaikka SQL –ja VBA-osaamisesta onkin apua. Suureksi avuksi Accessin käytössä ovat myös ohjatun luomisen toiminnot (engl. Wizard), joiden avulla lähes jokaisen Access-objektin luonti käy helposti.

Yhden tiedoston mallilla on hyvät ja huonot puolensa: varmuuskopioinnin ja versionhallinnan helppous, sekä tietokannan vaivaton jakaminen useille eri käyttäjille ja koneille. Huonona puolena tässä mallissa on tietokannan käyttämisen hidastuminen tiedoston koon paisuessa.

3.3 Muita tietokannan hallintajärjestelmiä

Tietokannan hallintajärjestelmiä on markkinoilla muitakin kuin Access. Jo mainittu Microsoftin toinen tuote MS SQL Server, joka on tarkoitettu ammattimaisempaan käyttöön kuin Access, on äänestetty useana perättäisenä vuotena parhaaksi relaatiotietokantojen hallintajärjestelmä - ratkaisuksi. Varustelutasoltaan samaa luokkaa kuin Access on Corelin Paradox. SQL serverin kovimpia kaupallisia kilpailijoita taas ovat Oracle Corporationin Oracle ja IBM:n DB2. Suosituimpia ilmaisia vaihtoehtoja ovat MySQL ja Borlandin Firebird. (SQL-opas: Johdatus SQL:n maailmaan, hakupäivä 11.11.2012.)

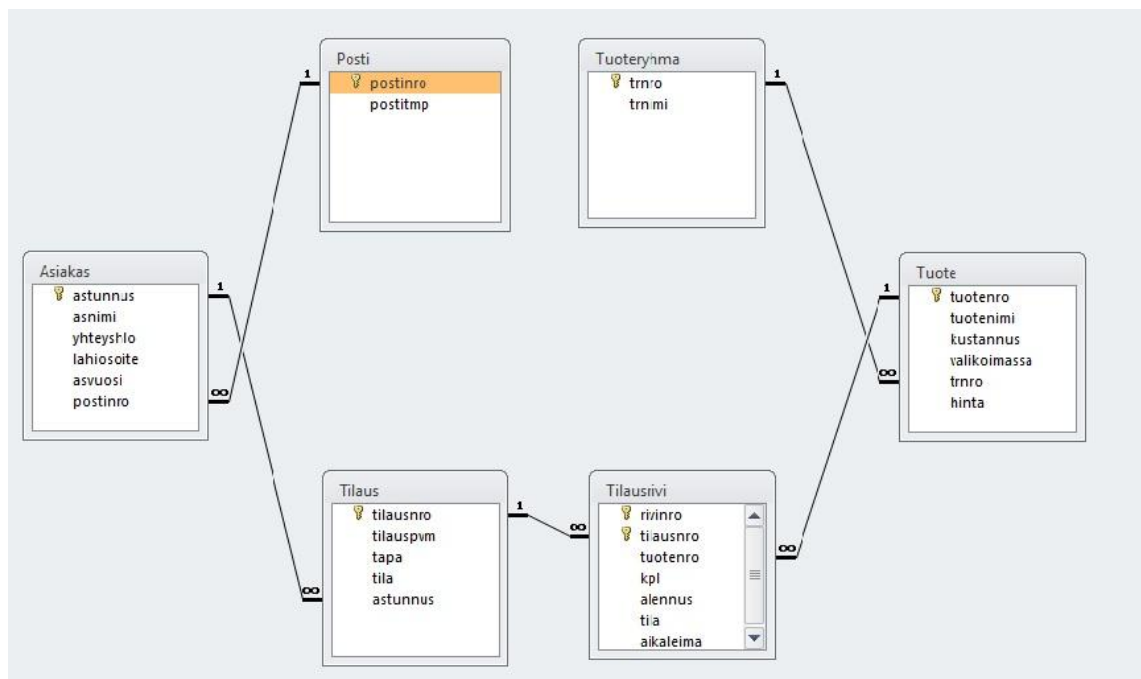
Eri tietokantojen hallintajärjestelmät eroavat toisistaan ominaisuuksiltaan, rajoituksiltaan ja hinnaltaan. Kaikkia hallintajärjestelmiä ei esimerkiksi voi ajaa Windows-ympäristössä, ja toiset ovat suunniteltu käytettäväksi Linux –ja Unix-ympäristöissä. Myös tietokannan koko asettaa omat rajoituksensa. Osasta tietokantoja puuttuu myös ominaisuuksia, kuten viite-eheyden säilymisen turvaaminen. (Wikipedia: Comparison of relational database management systems, hakupäivä: 11.11.2012.) Yhteistä edellämainituille tietokannan hallintajärjestelmille on kuitenkin se, että ne kaikki ymmärtävät SQL-kieltä, vaikka esimerkiksi Oracle hyödyntää tehokkuuden parantamiseksi

itse kehittämäänsä versiota SQL:sta, joka on kuitenkin rakenteeltaan ja komennoiltaan hyvin samanlainen SQL:n kanssa. (SQL-opas: Johdatus SQL:n maailmaan, hakupäivä 11.11.2012.)

4 DEMOX2 JA SEN TOTEUTUS

4.1 Alkuvalmistelut ja tietokanta

Aloitin työni tutustumalla muutamaaan Access-oppaaseen palauttaakseni mieleeni Accessin peruseriaatteet, samalla pitäen silmällä keinoja toteuttaa vaatimusmäärittelyssä esiinnousseet halutut toiminnalliset ominaisuudet. Seuraavaksi loin tietokannan tauluineen ja tietosisältöineen. Tietokannan rakenteeseen tehtiin työn eri vaiheissa useita muutoksia erinäisistä syistä johtuen. Kuvassa 4 on esillä tietokannan taulurakenteen lopullinen versio sarakkeineen ja yhteyksineen.



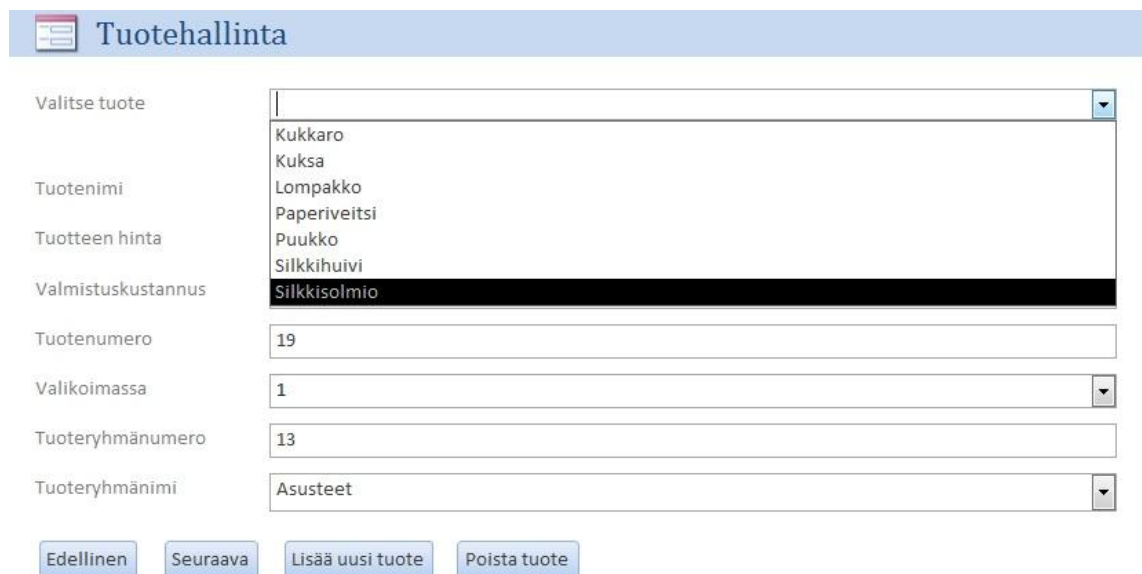
Kuva 4. Tietokannan taulut ja relaatiot - lopullinen versio

4.2 Lomakkeet ja SQL-kyselyt

Tietokannan ollessa valmis, seuraava askel oli käyttäjän ja tietokannan väliseen kommunikointiin tarvittavan käyttöliittymän toteuttaminen. Tätä työtä lähdin toteuttamaan kahdesta eri lähtökohdasta käsin: käytettävyyden ja oma oppiminen. Halusin oppia käyttämään tuoreinta Accessin versiota ja käyttää sovelluksen toteuttamiseen Accessia mahdollisimman monipuolisesti, kuitenkin käytettävyyden kärsimättä. Sovellusta tehtiin opetuskäyttöön,

käyttöliittymän tulisi siis luonnollisesti olla sellainen, ettei itse käyttäminen ja sen opettelu verottaisi resursseja varsinaisesta opetus -ja oppimistyöstä. Tämän toteuttamiseen sopii MacDonaldin (2010, 439) mukaan erinomaisesti lomakepohjainen lähestymistapa, jossa valikkolomakkeen kautta liikutaan muihin lomakkeisiin, sen sijaan että etsittäisiin objektit navigaatiopalkista. MacDonald lisää, että tämänkaltaisen järjestelmä on omiaan henkilöille, jotka eivät tunne Accessin saloja: jos navigaatiojärjestelmä on rakennettu oikein, käyttäjät voivat alkaa lisäämään tietokantaan tietoa ymmärtämättä Accessista yhtään mitään.

Päätin jakaa ohjelman toiminnot neljään kategoriaan, jako oli sekä looginen ja toisaalta taulurakenteen ohjaama: asiakkaat, tuotteet, tilaukset ja raportointi, jotka mahdollisesti kukin sisältäisivät omia alakategorioitaan, kuten esimerkiksi raportit asiakkaista, tuotteista ja tilauksista. Myöhemmin lisäsin kategorian postinumerojen, koska postinumerojen lisäystä oli vaikea upottaa asiakkaiden lisäyksen yhteyteen. Kuvassa 5 on esillä Tuotehallinta -lomake, alasvetovalikko josta valitaan haluttu tuote sekä painikkeet, joilla voidaan liikkua edelliseen ja seuraavaan tuotteeseen, poistaa valittu tuote tai siirtyä lomakkeelle, jolla lisätään uusi tuote tietokantaan.



Tuotehallinta

Valitse tuote

Tuotenimi

Tuotteen hinta

Valmistuskustannus

Tuotenumero

Valikoimassa

Tuoteryhmänumero

Tuoteryhmänimi


Edellinen Seuraava Lisää uusi tuote Poista tuote

Kuva 5. Tuotehallinta -lomake

Lomakepohjainen tietokannan hallinta ei ole ainoa mahdollinen ratkaisu, päivittäiset tietokannan ylläpitämiseen liittyvät toiminnot pystyy suorittamaan myös taulunäkymässä (MacDonald 2010, 367). Tämä tosin on huomattavan työlästä, varsinkin jos tietokannan kanssa joutuu olemaan tekemisissä päivittäin, ja etenkin mikäli samaan aikaan käsiteltävä tieto on jaoteltu useampaan eri tauluun (esimerkiksi tuotteet ja tuoteryhmät - kaksi eri taulua). MacDonald (2010, 368) mainitsee

neljä asiaa, joilla lomakkeet päihittävät taulunäkymän tietokannan hallinomisessa. Ensinnäkin tieto on paremmin järjesteltävissä näkyville etenkin isojen taulujen ollessa kyseessä. Toiseksi lomakkeeseen saa näkyviin mitä tahansa tekstiä, siis muutakin kuin mitä taulussa lukee. Kolmantena toisiinsa liittyvien taulujen käsittely on huomattavasti helpompaa ja neljänneksi tauluihin saa kontrolleja, esimerkiksi painikkeita, joilla voi suorittaa monenlaisia tehtäviä.

Edellisen valossa lomakkeet olivat ainoa ratkaisu käyttöliittymän toteuttamiseen. Aloin siis tehdä lomakkeita yksi kerrallaan, lopulliseen versioon niitä tuli 12 kappaletta mukaan lukien navigaatiolomake. Accessilla lomakkeiden luonti pelkästään taulujen pohjalta käy muutamalla hiiren napin painalluksella. Pienellä säätämällä myös useista eri tauluista tietonsa ottavat lomakkeet saa toimimaan ainakin tiettyyn rajaan asti. Lomakkeiden käydessä yhä monimutkaisemmiksi huomasin, että on parempi rakentaa itse SQL-kyselyt, joiden pohjalta lomakkeet luodaan. Ilman toteuttamaani tilausjärjestelmää tätä ei olisi tarvinnut tehdä. Tilauslomake alilomakeineen tarvitsi kuitenkin tietoa niin monesta taulusta, sen piti pystyä tallentamaan tietoa lähes yhtä moneen tauluun, sekä tehdä laskutoimituksia käyttäjän antaman tiedon perusteella, että ilman järkeviä SQL-kyselyjä uusien tilausten tallentaminen kantaan lomakkeella ei olisi taidoillani onnistunut. SQL-kieli sillä tasolla, jota tässä työssä vaadittiin, oli onneksi varsin nopea oppia SQL:n perusteet joskus opetelleena. Kuvassa 6 on esillä tilauslomake sekä alilomake, jolla valitaan tilattavat tuotteet.

 **Lisää uusi tilaus**

Valitse tilaaja

Näsi Oy

tilauspäivämäärä

26.11.2012

tilaustapa

V

L = manuaalisella lomakkeella V = verkon kautta

tilauksen status

M

tyhjä = tilaus kirjattu, M = maksettu, L = laskutettu, T = toimitettu, O = osittain toimitettu

Valitse tilattavat tuotteet

Tuotenimi	Hinta	Kappalemäärä	Alennus%	Rivihinta	Tila
Kuksa	28	10	10	252,00 €	M
*					

Record: 1 of 1

No Filter

Search

Edellinen

Seuraava

Sulje

Kuva 6. Uusi tilaus –lomake

Kun lomakkeet sisäisine navigaatiojärjestelmineen olivat kutakuinkin kunnossa, oli aika luoda niille yhteinen navigaatiolomake, eräänlainen päävalikko. Yksi Access 2010:n uudistuksista vanhempaan 2007 versioon nähden on navigaatiolomake. Navigaatiolomakkeen luonti Accessilla on hyvin yksinkertaista, ja sen täyttäminen halutuilla lomakkeilla onnistuu kätevästi drag and drop -menetelmällä. Kuvassa 7 näkyy toteuttamani navigaatiolomake, jossa lomakkeet ovat jaoteltu viidelle eri välilehdelle. Kuvassa on mustalla värillä rajattu navigaatiolomakkeen mahdollistama välilehtivalikko. Joidenkin välilehtien alta löytyy useita lomakkeita. Kuvan Asiakkaat välilehden alla ei ole muita kuin Asiakashallinta. Vasemmalla auki myöskin mustalla värillä rajattuna Accessin oma navigaatiopalkki, josta löytyy samat lomakkeet kuin navigaatiolomakkeesta.

The screenshot displays the Microsoft Access interface. On the left is the 'Navigation Pane' (Accessin oma navigaatiopalkki) with a search bar and a list of objects under the 'Forms' category: asiakashallinta_form, navi_form, postihallinta_form, tilaus_subform, tilaushallinta_form, tuotehallinta_form, tuoteryhmähallinta_form, uusi_asiakas_form, uusi_postitmp_form, uusi_tilaus_form, uusi_tuote_form, and uusi_tuoteryhmä_form. The main window is titled 'Demox 2' and contains a navigation form with five tabs: 'Asiakkaat', 'Tuotteet', 'Tilaukset', 'Postinumerot', and 'Raportointi'. The 'Asiakkaat' tab is selected, showing a form titled 'Asiakashallinta'. This form has a 'Valitse asiakas' dropdown at the top. Below it are several text input fields: 'Asiakastunnus' (containing 'SARA'), 'Asiakkaan nimi' (containing 'Sara Oy'), 'Yhteyshenkilö' (containing 'Herva'), 'Lähiosoite' (containing 'Runeberginkatu 2'), 'Asiakk. alkamisvuosi' (containing '2009'), 'Postinumero' (containing '00100'), and 'Postitoimipaikka' (containing 'Helsinki'). At the bottom of the form are four buttons: 'Edellinen', 'Seuraava', 'Poista asiakas', and 'Lisää uusi asiakas'.

Kuva 7. Navigaatiolomake

4.3 Raportit

Tuoreiden SQL-kyselyiden pohjalle oli helppo luoda raportit. Päädyin toteuttamaan neljä eri raporttia: yhden asiakkaille, toisen tilauksille ja kaksi tuotteille. Accessin suodattimien avulla raporteista saa paljon erilaista tietoa irti, kunhan vain tarvittava tieto on sisällytettynä raportin SQL-kyselyyn. Suodattaminen Accessin tapauksessa tarkoittaa tietojen hakemista raportista käyttäjän määrittämien ehtojen perusteella, esimerkiksi kaikkien tuotteiden tulostamisen sijaan voi hakea vain tiettyyn tuoteryhmään kuuluvat tuotteet. Vaihtoehtona muutamalle suodattimin

muokattavalle raportille, olisi tehdä SQL-kyselyt jokaista raportointia vaativaa tietoa varten. Raportoinnissa tarvittavia tauluja tietokannassa oli kuitenkin verrattain vähän, ja suodattimien ollessa varsin helppokäyttöisiä, päädyin suodatusmenetelmään useiden erilaisten SQL-kyselyiden sijaan. Tämän tekniikan etu MacDonaldin (2010, 329) mukaan on, että suodatinasetuksia voi vaihdella nopeasti peräkkäin. Hän lisääkin, että mikäli käyttää yhtä raporttipohjaa monia erilaisia tulosteita varten, tämä lähestymistapa on paras. Kuvassa 8 näkyy raportti tietokantaan tallennetuista tuotteista.

Tuoteluettelo		
Tuote	Tuoteryhmä	Hinta
Silkkihuivi	Asusteet	20,00 €
Silkkiolmio	Asusteet	15,00 €
Kukkaro	Laukut	16,00 €
Lompakko	Laukut	31,00 €
Puukko	Talous	27,00 €
Kuksa	Talous	28,00 €
Paperiveitsi	Toimisto	16,00 €

26. marraskuuta 2012

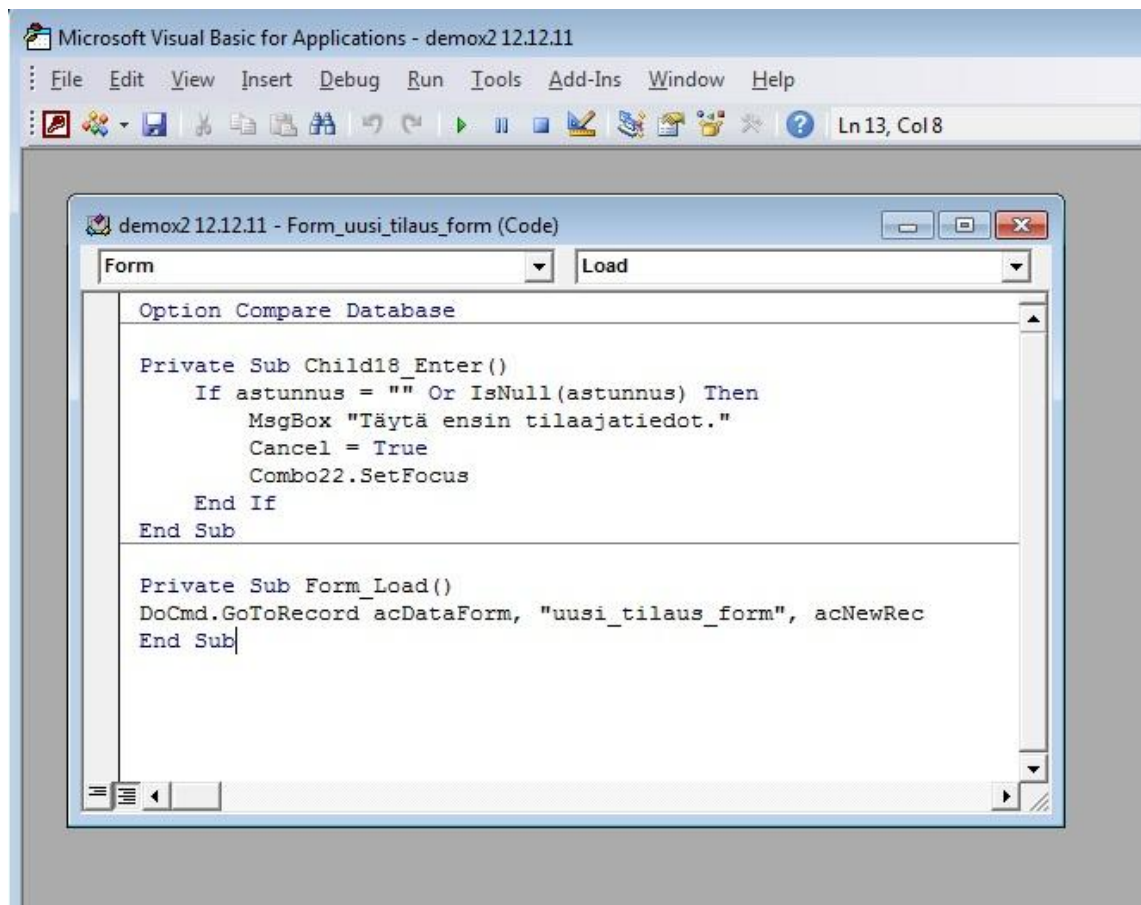
Page 1 of 1

Kuva 8. Raportti tietokannassa olevista tuotteista

4.4 Visual Basic for Applications

Lisää toiminnallisuutta lomakkeisiin halutessani huomasin, että joudun opettelemaan myös Visual Basic for Applications (VBA) ohjelmointikieltä voidakseni toteuttaa haluamani ominaisuudet. Näitä olivat mm. lomakkeen siirtyminen automaattisesti tyhjiin tietueeseen tietojen syöttämistä varten ja mahdollisuus hallita lomakkeen kenttien täyttämisyjärjestystä. Myöhemmin olen huomannut, että ensinmainittu olisi kaikesti onnistunut Accessin sisäänrakennetuilla makroilla, mutta tähän tarkoitukseen soveltuvaa makroa en syystä tai toisesta löytänyt sitä tarvitessani. Jälkeenpäin ajatellen hyvä niin, koska VBA mahdollistaa makroja suuremman toiminnallisuuden: "Macros are plenty of fun, but they can do only so much. If you can't find a readymade macro action that does the job you want, you can't use a macro. No such limit applies in the world of Visual Basic code, where you can do just about anything." (MacDonald 2010, 535.) VBA-kieli tai muutkaan samankaltaiset kielet eivät olleet minulle entuudestaan tuttuja ennen tätä työtä. Yrityksen, erehdyksen ja Internetin Access-forumien viestejä seuraamalla sain kuitenkin haluamani ominaisuudet toteutettua. Kuvan 9 VBA-kielillä kirjoitettu koodi saa lomakkeen menemään

tyhjään tietueeseen tietojen syöttämistä varten sekä estää tilauksen tekemisen, mikäli asiakasta ei ole valittu, ja vie hiiren cursorin Valitse tilaaja –pudotusvalikon kohdalle (kuva 10.)



Kuva 9. VBA:lla kirjoitettua koodia.

Valitse tilaaja

tilauspäivämäärä

tilaustapa

L = manuaalisella lomakkeella V = verkon kautta

tilauksen status

tyhjä = tilaus kirjattu, T = toimitettu, O = osittain toimitettu

Valitse tilattavat tuotteet

Tuotenimi	Hinta	Kappalemäärä	Tila
*			

Record: 1 of 1 No Filter Search

Microsoft Access

Täytä ensin tilaajatiedot.

OK

Kuva 10. Virheilmoitus

4.5 Viimeistely

Toimeksiantajan testauksessa sovelluksesta tavattiin virheitä ja toiminnallisia epäloogisuuksia, jotka olivat oman seulan välttäneet. Näiden korjaamisen ja toiminnallisuuden parantamisen lisäksi käytin ehkäpä turhan paljon aikaa yrittäessäni saada tilauslomakkeelle tilauksen kokonaissummaa näkymään – yksinkertainen VBA:lla kirjoitettu rivi koodia, joka ei kuitenkaan toiminut. Access antoi parhaassa tapauksessa laskutoimituksen suorittamisesta virheilmoitukseksi vain moniselitteisen #Error viestin; tämä ominaisuus jäi toteuttamatta.

Kun työ alkoi todellisuudessa olla loppusuoralla, saatoin tietokannan tietosisällön lopulliseen muotoonsa, poistin tietokannasta turhat tiedot, lomakkeet ja muut objektit sekä siistin ulkoasua, objektien nimiä ja oikeinkirjoitusta. Tässä vaiheessa myös kirjoitin lopulliseen muotoonsa opiskelijoille tulevat tehtävät (Liite 1) sekä ohjeen sovellukseni ja Accessin joidenkin ominaisuuksien käytöstä (Liite 2.)

5 YHTEENVETO JA POHDINTA

Opinnäytetyöni tavoitteena oli toteuttaa tietokantasovellus Access 2010 -ohjelmalla aikataululla, jolla sen saisi käyttöön talvilukukaudelle 2012 Tietokantojen perusteet –kurssin oppitunneille. Alkuperäisen aikataulun mukaan valmiin sovelluksen käyttöönottestaus suunniteltiin tehtäväksi vuoden 2011 syksyllä. Aikataulu kuitenkin petti suurelta osin siksi, että arvioin työmäärän huomattavasti vähäisemmäksi, kuin mitä se todellisuudessa tuli olemaan ja täten varasin liian vähän aikaa sovelluksen toteuttamiseen. Testauksen väliin jättämisestä huolimatta, toimitin valmiin sovelluksen oheisdokumentteineen (liitteenä) toimeksiantajalle joulukuun puolella välissä ja sovellus otettiin suunnitellusti käyttöön oppitunnilla.

Eniten aikaani vei lomakkeiden toiminnallisuuden toteuttaminen, jonka parissa myös kohtasin suurimmat ongelmat. Uuden ominaisuuden toteuttaminen eteni usein seuraavanlaista kaavaa noudattaen. Aluksi on selvillä minkälainen toiminto täytyy toteuttaa, seuraavaksi selvitetään mahdollisia vaihtoehtoja sen toteuttamiseksi, valitaan yksi, opetellaan sen käyttö ja lopuksi todetaan ettei toteuttaminen näin onnistukaan ja aloitetaan alusta. Yleensä etsin ongelmiini apua Internetin Access –forumeilta, joiden neuvoja seurattessani kävi usein ilmi, että kyseinen toiminto kyllä onnistuu Access 2007:lla, mutta ei 2010:lla. Access 2010 oli tietokantasovelluksen toteuttamisen aikoihin vielä varsin tuore tapaus, jonka vuoksi on helposti ymmärrettävissä, minkä takia Internetin forumeilta ei aina apua löytynyt.

Toinen ongelma oli tietokantasovelluksen tai tietokannan ajoittainen korruptoituminen, josta Access ilmoitti yleensä virheilmoituksella: "A problem occurred while Microsoft Access was communicating with the OLE server or ActiveX Control." Virheilmoituksen jälkeen yleensä esimerkiksi macrot eivät toimineet ollenkaan tai johonkin tietokannan osaan tai toimintoon ei päässyt käsiksi. Joskus virheilmoitus katosi uudeelleenkäynnistämällä sovelluksen tai nimeämällä .accdb –tiedoston uudelleen, mutta usein joutui palaamaan takaisin vanhempaan versioon. Muutaman korruptoitumiskokemuksen jälkeen otin käyttöön versionhallinnan ulkoisella kovalevyllä, joka tulikin tarpeeseen, tietokannan korruptoituminen oli ikävän yleinen riesa. Työtä tehdessäni luulin tietokannan ajoittaista korruptoitumista Accessin virheeksi – Internetin Access –forumeita selatessani kävi ilmi, etten ollut ainoa tämän ongelman kanssa kamppaileva. Oppitunnilla ei kuitenkaan korruptoitumisongelmia havaittu, ja vastikään kävi ilmi, että PC:n, jolla tietokantasovellusta toteutin, kovalevy oli hajoamaisillaan. Tein tietokantasovellusta ainoastaan

tällä yhdellä ja samalla PC:lla, joten on syytä olettaa, että korruptoituminen johtui rikkinäisen kovalevyn tekemistä kirjoitus –ja lukuvirheistä.

Työni lopputulokseen olen jokseenkin tyytyväinen. Vaatimusmäärittelyyn vastattiin ja sain haalittua itselleni suuren määrän Access –osaamista. Joitakin asioita, kuten sovelluksen graafisen puolen, olisi voinut toteuttaa paljon paremminkin. Sovelluksen ulkonäkö oli kuitenkin alusta asti hyvin alhainen prioriteetti, kuvankäsittely ja grafiikka eivät ole koskaan ollut vahvoja puoliani. Käytettävyys eli tässä tapauksessa lomakkeiden toiminnot, on myös asia, jota olisi voinut hioa ja parantaa paljonkin, mutta aikataulu ei kuitenkaan antanut enää periksi. Onnistuneeseen lopputulokseen vaikutti omalta osaltaan suuresti hyvin tiivis yhteistyö ja ohjaus opinnäytetyön ohjaajan kanssa, jota ilman tämä työ ei olisi valmistunut.

LÄHTEET

Codd, E. F. 1970. A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, Vol. 13, No. 6.

Hovi, A. & Huotari, J. 2005. Tietokantojen suunnittelu ja indeksointi. 1. painos. Jyväskylä. Docendo Finland Oy.

Keinonen, K. 2010. Access 2010 – Edistynyt käyttö. Suomi: Ornanet

Laakkonen, M. & Lamminpää, S. & Malaprade, J. (toim.) Informaatioteknologian filosofia. Jari Palomäki: Käsitteet ja käsittemallit informaatiojärjestelmissä. Tampere: Juvenes Print

Lambert, S. 2008. Access 2007 Tehokas hallinta. Helsinki: Readme.fi.

MacDonald, M 2010. Access 2010 – The missing manual. United States: O'Reilly Media / Pogue Press

OAMK – luonnontieteiden ala (tietojenkäsittely). Hakupäivä 4.11.2012.
http://www.oamk.fi/koulutus_ja_hakeminen/nuoret_suomenkielinen/luonnontieteet/index.php?sivu=tietojenkäsittely

Ruuhonen, M. & Salmela, H. 2005. Yrityksen tietohallinto. Helsinki: Edita Prima Oy

SQL-Opas: Johdatus SQL:n maailmaan. Hakupäivä 11.11.2012.
<http://www.2kmediat.com/sql/alkeet.asp>

Turun kauppakorkeakoulun sidosryhmälehti Mercurius. Tietojärjestelmiin panostettava riittävästi.
Hakupäivä 28.10.2012.
<http://www.mercurius.fi/index.php?page=3b3907b29e0f395b3b01065bbc9fdee>

Ullman, J. & Widom, J. 1997. A First Course in Database Systems. Prentice Hall Engineering/Science/Mathematics

Wikipedia. Codd's 12 rules. Hakupäivä 20.12.2012. http://en.wikipedia.org/wiki/Codd's_12_rules

Wikipedia. Comparison of relational database management systems. Hakupäivä 11.11.2012.

http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems

Tehtäviä opiskelijoille

Tuotehallinta

1. Tutki miltä kannan sisältö näyttää. Ota Print Screen kuva kannan tauluista tuote ja tuoteryhmä.
2. Lisää yrityksen tuotteet (3 eri tuoteryhmää ja niihin useampia tuotteita) tietoineen tietokantaan.
3. Hinnat muuttuvat. Muuta parin tuotteesi hintaa.

Asiakas -ja postihallinta

1. Lisää uusi postitoimipaikka ja postinumero kantaan.
2. Lisää asiakas, jonka osoite on aiemmin lisäämälläsi paikkakunnalla.
3. Eräs asiakas ilmoittaa uuden yhteyshenkilön nimen, korjaa tieto kantaan.
4. Asiakas muuttaa, muuta osoitetta.

Tilaushallinta

1. Anna 3-4 kaverillesi tuoteluettelosi ja tilauslomake. Pyydä heitä tekemään tilaus.
2. Lisää uudet asiakkaat tietokantaan.
3. Tallenna saamasi tilaukset tietokantaan.
4. Tee vielä itse muutama tilaus parille eri asiakkaalle. Tee ainakin yhdelle asiakkaalle 2 eri tilausta.
5. 'Toimita' parin asiakkaan tilaukset täydellisinä. Jokin tavara loppuu varastosta, toimita jokin tilaus niin, että vain osa tuotteista voidaan toimittaa.
6. Eräs asiakas muuttaa tilaustaan. Lisätään tilaukseen uusi tuote tietoineen.
7. Eräs asiakas peruu yhden tilaamistaan tuotteista. Poista tuote tilauksesta.

Raportointi

1. Aja lista asiakkaista, jotka asuvat Oulussa.
2. Listaa tuotteet, joita ei ole valikoimassa tällä hetkellä.
3. Aja lista tilauksista, jotka on tilattu viimeisen viikon sisällä.
4. Aja lista tilauksista, jotka ovat osittain toimitettu.

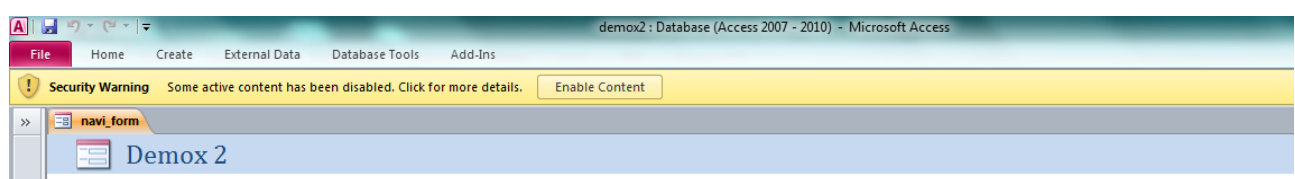
Yhteenveto (vertailu Demox ja Demox2 toiminnallisuus):

1. Kuvaa, mitä erilaista toiminnallisuutta Demox2 mahdollistaa verrattuna Demoxiin?
2. Mitä muutoksia Demox2 kantaan tarvittaisiin, jos halutaan seurata tuotteiden määrää varastossa?

Yleisiä ohjeita Accessin ja Demox2:n käyttöön.

Ensimmäinen käynnistys

Kun tietokanta avataan ensimmäisen kerran, antaa Access tietoturvarvoroituksen, joka pyytää sallimaan aktiivisen sisällön. Tämä tarkoittaa tietokantaan ympätyjä koodin pätkiä, makroja, SQL-lauseita jne. Painamalla Enable Content sallitaan koodin ajaminen. Mikäli ohitit Microsoftin ylireagoivan varoitusviestin painamalla raksia, käynnistä sovellus uudelleen ja salli sisältö.



Sovelluksen käynnistyessä avautuu navigaatiolomake, joka on uusi ominaisuus Access 2010:ssa. Kaikki navigaatiolomakkeen alta löytyvät lomakkeet ja raportit, sekä itse taulut, joihin lomakkeiden tiedot perustuvat, löytyvät myös vasemmalta avautuvan Navigation Pane:n alta. Navigaatiolomake alilomakkeineen on tarkoitettu tietokannan "käyttöliittymäksi."



Vasemmallalla navigation pane avattuna.

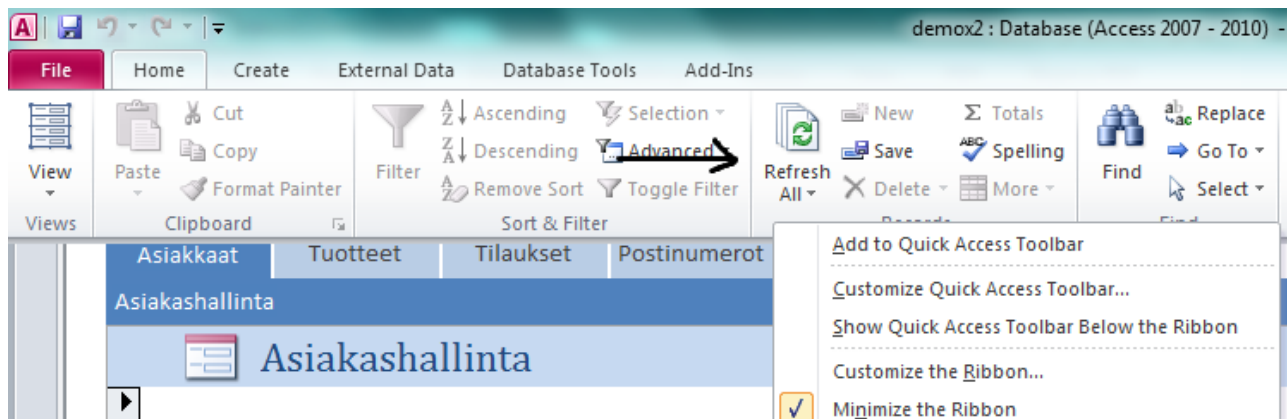
Demoxin käyttö

Tietuiden lisääminen

Kun lisää tietoa lomakkeiden kautta, muutoksia ei tarvitse erikseen tallentaa. Sama pätee tietojen muokkaamiseen ja poistamiseen. Yleensä riittää, että täyttää vaadittavan kentän, ja siirtyy esim. tabulaattorilla seuraavaan kohtaan. Osa lomakkeilla olevista kentistä ovat autonumber-tyyppisiä, jotka eivät vaadi käyttäjältä muuta kuin sen että kenttä kerran aktivoidaan, jolloin Access syöttää kenttään itse generoimansa numeron. Tämän takia lomakkeilla on kätevä siirtyä tabulaattorilla kentästä toiseen.

Lomakkeiden päivittäminen

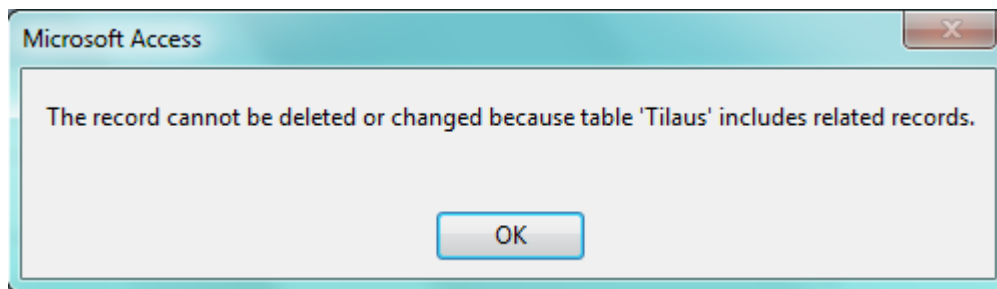
Kun lisää dataa tietokantaan, ja palaat tarkastelemaan niitä päälomakkeeseen, lomake täytyy yleensä päivittää (ts. ajaa query, johon lomakkeen tiedot perustuvat, uudelleen). Accessin oma, avoinna olevien lomakkeiden, queryjen jne. päivitys tapahtuu 60 sekunnin välein. Päivitysväliä voi muokata lyhemmäksi, mutta luonnollisesti se hidastaa tietokannan toimintaa. Tietojen lisäämisen jälkeen päälomakkeelle palattaessa kannattaa yleensä tehdä manuaalinen päivitys joka löytyy valintanauhasta (ribbon) kohdasta Home-> Records -> Refresh all, tai pelkkä Refresh mikäli haluat päivittää vain sillä hetkellä valitun lomakkeen. Valintanauha kannattaakin päivityksiä varten asettaa pysyvästi näkyviin seuraavasti: oikea hiiren painallus valintanauhan päällä ja rasti pois kohdasta Minimize the Ribbon.



Tietueiden poistaminen

Joskus yritettäessä poistaa tietuetta saattaa Access antaa seuraavanlaisen virheilmoituksen: "The record cannot be changed or delete because table 'tähän taulun nimi' includes related records." Tämä tarkoittaa esimerkin kautta sitä, että jos koitat poistaa tuotetta, joka on lisätty johonkin tilaukseen, ei tuotetta voi poistaa, ennen kuin se poistetaan kaikista tilauksista. Virheen

voi ohittaa painamalla "OK" ja sulkemalla raksista seuraavaksi avautuvan Macro Single Step ikkuna .



Raportit

Raporteista saa kaveittua haluamansa tiedon käyttämällä suodattimia. Valitse tietue, jonka perusteella haluat raportin suodattaa (esim. Tilaukset -raportti ja asiakkaan nimi MasiYards. Oikealla hiiren napilla avautuu valikko, josta (Text/Number/Date) Filters kohdan alta, voi raporttiin lisätä erilaisia suodattimia, riippuen tietueen tietotyypistä. Suodattimia voi olla useita päällä yhtä aikaa. Kun poistut raportista, suodattimet katoavat.

Asiakkaat	Tuotteet	Tilaukset	Postinumerot	Raportointi
Asiakkaat	Tilaukset	Tuotteet	Tuotelu	
Tilaukset				
Tilausnumero	Tilauspäivä	As. tunnus	Nimi	
22	15.9.2011	ITBO	IT-De	
17	19.9.			
18	11.11.			
20	10.2.			
24	20.10.			
21	13.4.			
23	10.10.			
19	12.2.			
11. joulukuuta 2011				

Cut

Copy

Paste

Insert

Merge/Split

Sort Oldest to Newest

Sort Newest to Oldest

Clear filter from tilauspvm

Date Filters

Equals 15.9.2011

Does Not Equal 15.9.2011

On or Before 15.9.2011

Equals...

Does Not Equal...

Before...

After...

Between...

Tomorrow

Today

Yesterday

Next Week

This Week

Last Week

Next Month

This Month

Last Month

Next Quarter

This Quarter

Tapa	Tila
L	T
L	M
V	M
V	M
L	
V	L
V	
L	M

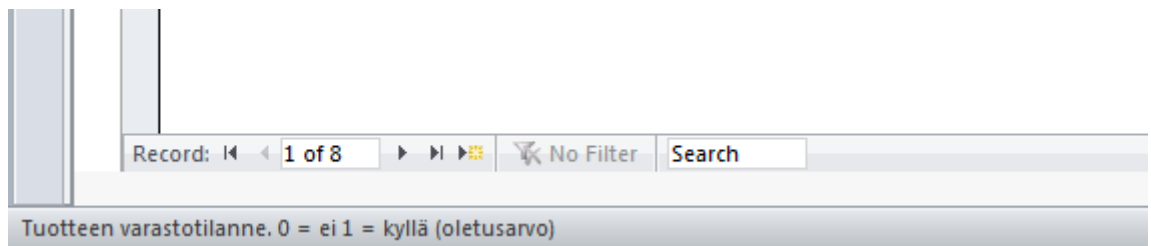
Page 1 of 1

Date –tietotyypin suodatusmahdollisuudet

Muuta

Status Bar

Ominaisuus joka jää helposti huomaamatta, Accessin käyttöliittymän alareunasta löytyvää status baria kannattaa tarkkailla mikäli on epäselvyyksiä vaikkapa kenttien ja niihin täytettävän datan merkityksestä ja muodosta. Esimerkiksi tuote -taulua tarkastellessa, kun valikoimassa -tietue on valittuna, kertoo status bar että numero 1 tarkoittaa tuotetta olevan valikoimassa ja 0 että sitä ei ole.



Virhetilanteet

"A problem occurred while Microsoft Access was communicating with the OLE server or ActiveX Control." Tämä virhe on häirinnyt työskentelyä silloin tällöin, tarkkaa syytä virheilmoitukseen ei ole tiedossa, yleensä virhe poistuu käynnistämällä sovellus uudelleen. Joskus tämä johtuu siitä että Access-tietokanta on syystä tai toisesta enemmän tai vähemmän korruptoitunut ja se voi estää tiettyjen macrojen toiminnan kokonaan. Mikäli virhe toistuu kannattaa yrittää nimenä sovellus (.accdb tiedosto) uudelleen Windowsissa tai poistaa tiedosto ja kopioida se uudestaan koneelle.